SE12-3
(1975. 11. 14)

ワーキング・セット・モデルの特性測定

小野隆喜，中崎良成，箱崎勝也
（日本電気株式会社 中央研究所）

1．はじめに

マルチプログラミング・システムにおいて，仮想記憶方式を採用したシステムが普及しつつある。このようなシステムでは，メモリ管理，特にメモリの使用効率の如何によりシステム全体の処理効率は大きく影響される。

種々のメモリ管理方式の下でメモリの使用効率を観察する場合，プログラムのメモリ参照特性を把握することが重要になる。通常，メモリの参照はプログラム構造に依存する訳であるが，一般的には小さなループ中を実行し，適当な間隔で大きなループへジャンプし，稀れに更に異なる番地にジャンプするものと考えられる。この時，ループ中を実行する，即ちメモリ参照がある領域に集中することになる。このメモリ参照の集中をプログラムのローカリティと呼んでいる。プログラムのローカリティを扱ったモデルの一つに，ワーキング・セットの概念がある。このモデルでは，メモリ管理を動的に行なうことに特徴がある。即ち，メモリ参照の過去の履歴を基に近い将来の参照を予測して管理しようとするものである。

ワーキング・セット・モデルに則して実際のデータを測定[1][2]し，性能を解析した報告[3]はまだ少ない。

そこで今回，我々はメモリ管理の性能へ与える影響を検討するに先立って，ワーキング・セット・モデルで定義された種々のパラメータについてデータを測定した。以下にこのモデルについて説明し，次にデータの測定方法と，得られた結果の一部を記述し，最後に考察を述べる。

2．ワーキング・セット・モデル[4]

ワーキング・セット・モデルは，P·J·Denningによって提案されたモデルである。あるプログラムのワーキング・セットとは，そのプログラムがある時刻 t からそれ以前の τ で時間の間に参照した情報の集合（$W(t,\tau)$）として定義され，ここで特に τ をウィンド・サイズと言う。以下にワーキング・セット・モデルで定義される主なパラメータについて述べる。

ワーキング・セット・サイズ：$w(t,\tau)$

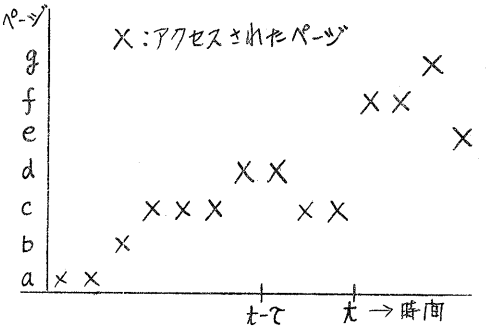
ワーキング・セット $W(t,\tau)$ に含まれるページ数を表わす。

平均ワーキング・セット・サイズ：$\overline{w(\tau)}$

$w(t,\tau)$ より

$$\overline{w(\tau)} = \lim_{K\to\infty} \frac{1}{K} \sum_{t=1}^{K} w(t,\tau)$$

で定義される。ここで $t<0$ においては，$w(t,\tau)=0$，メモリ・アクセスは無限に続くものと考える。



第1図．ページアクセス例

ミッシング・ページ率：$m(\tau)$

第1回において $W(t,\tau)$ のページを確保して $(t, t+1)$ の間処理を行うとページ f についてミッシング・ページが起きる。ミッシング・ページ率は，$\overline{w(\tau)}$ のページを確保して次のメモリ・アクセスを行った時，ミッシング・ページが生じる確率であり，次の式で表わすことができる。

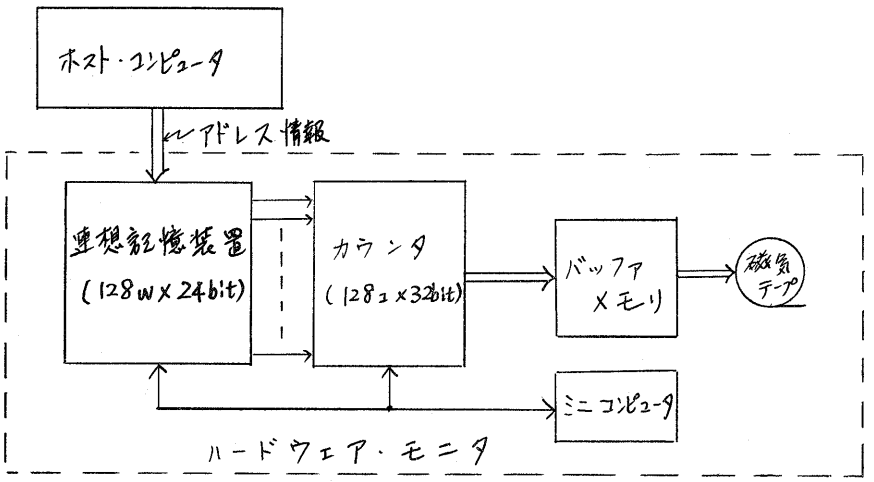
$$m(\tau) = \lim_{K\to\infty} \frac{1}{K} \sum_{t=1}^{K} \left\{ w(t+1, \tau+1) - w(t, \tau) \right\}$$

このようなワーキング・セット特性を定量的に求めることによって，プログラムの局所性に関する動特性が把握できる。それらの特性から適当なワーキング・セットを主記憶上に確保してプログラムを実行することにより，スラッシングを防止でき，主記憶を有効に各プログラムに割り当てるために最適なプログラム多重度を決定する上で有用な情報となる。また，プログラムの局所性の評価のために用いることもできる。

3．実験

(1) 測定方法

データ測定方法の概要を第2図に示す。

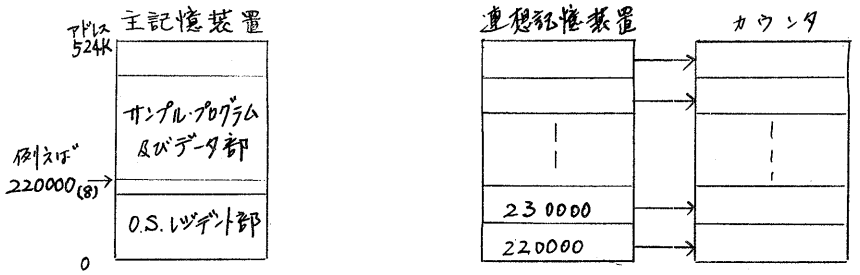


第2図．データ測定の概要図

ホストコンピュータには，NEAC 2200 シリーズ，モデル500（主記憶 524 Kch，サイクル・タイム 1.5 μs），ミニコンピュータは NEAC-M4 を使用している。データ測定装置は，先に当所で試作したハードウェア・モニタ（SYDAS）[5] の機能の一部を用いている。まずホスト・コンピュータ上でサンプル・プログラムを実行する。その時のアドレス情報を電気的に測定装置へ取り出す。測定装置では，予めミニコンピュータにより連想記憶装置の各ワードにページを表わすアドレス値をセットしておき，このアドレス値とホスト・コンピュータからのアドレス情報をこの連想記憶装置で比較する。そこで両アドレスが一致すると，連想記憶装置の各ワードに対応したカウンタで計数する。このカウンタの計数値，即ちページのアクセス回数値を一定時間間隔毎にバッファ・メモリを介して磁気テープに記録する。解析は，この磁気テープに記録されたデータを基に

バッチ処理で行う。

第3図にホスト・コンピュータの主記憶装置と，ハードウェア・モニタの連想記憶装置の関連を示す。



第3図．主記憶装置と連想記憶装置

第3図において，例えばサンプル・プログラムとそのデータをホスト・コンピュータの主記憶装置の220000(8)番地からロードする。この時ページ・サイズを4Kchとすると，連想記憶装置の各ワードには，順次8進数で220000，230000，240000，--- をセットする。そして実際に連想記憶装置で一致を検出する時は装置で下位12ビットについて注目しないように制御する。このようにしてホストコンピュータ稼動時のアドレス・レジスタの内容と，連想記憶装置に予めセットした値（この値はその制御によりページ・アドレスと考えられる）と比較し，各ページに対するアクセス回数をカウンタで計数する。故に，仮想的にページング方式を導入し，各ページに対するアクセス回数とほぼ等価な測定が行なえる。

測定において次に考慮したことは，データロストの防止である。カウンタの計数値を一定時間間隔（プロセス・タイムにおいて）毎に磁気テープに記録するがこの時間間隔を短かくするとデータロストが生じる。このために，ハードウェアモニタ側からホスト・コンピュータの動作，即ち停止及び再スタートを制御することができるようにしてある。

(2) サンプル・プログラム

この実験では，測定上の制限からオーバレイを行なわず，入手し易いユーザのプログラムを任意に選択してサンプル・プログラムとした。現在までに解析を終えたプログラムは，FORTRAN及びCOBOLで記述したもの，ユーティリティ・プログラムとしてソートのフェーズを単位としたものがある。

FORTRAN 1 ：逐行列計算．ソース・カード枚数 501枚
FORTRAN 2 ：数値計算．ソース・カード枚数 187枚
COBOL ：事務計算
SORT ：プリソート部．

(3) 解析項目

ワーキングセットモデルについては，2章で述べたが，SYDASの測定特性を考慮して作成された解析プログラムにおける解析項目の求め方について説明する。SYDASでの測定時間単位（サンプリングタイム）をUmsとする。測定データとしては，U＝1ms，10msがある。

平均ワーキングセットサイズ　$w(\tau)$

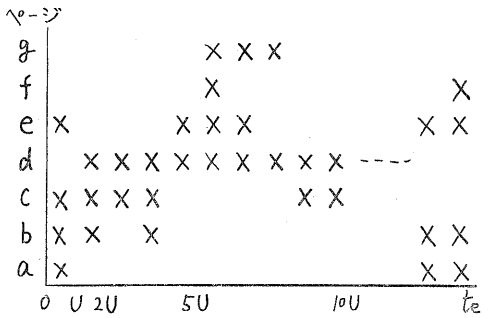
$w(\tau)$は，プログラム実行中のある時刻tにおいてウィンドサイズτとした時のワーキングセットサイズ$w(t,\tau)$を　プログラム実行開始時($t=0$)から，Um

s 刻みにプログラム終了時($t=t_e$)まで求め、それらのワーキングセットサイズの和を プログラム開始時から終了時までをU ms 刻みにした数(t_e/U) で割った値である。

$$W(\tau)=\frac{\sum_{n=1}^{[t_e/U]} W(nU,\tau)}{[t_e/U]}$$

[ ]：ガウス記号

$W(\tau)$ は、ウインドサイズをτとして、時刻 t が進むに従って変化する $w(t,\tau)$ の平均値である。解析結果より、ウインドサイズを変化させた場合の平均ワーキングセットサイズの変化を知ることができる。



オ4図 SYDASで測定する場合のメモリアクセス

ミッシングページサイズ $m(\tau)$

プログラム実行中のある時刻 t においてウインドサイズをτとした時に $W(t,\tau)$ をメモリに確保すると仮定して、次の U ms 中にページフォルトが生ずる回数を $M(t,\tau)$ とする。$m(\tau)$は、$M(t,\tau)$をU ms 刻みで、プログラム実行開始($t=0$)から終了時($t=t_e$)まで求め、それらのページフォルト回数の和を、開始時から終了時までのサンプリング回数(t_e/U) で割った値である。

$$m(\tau)=\frac{\sum_{n=0}^{[t_e/U]-1} M(\tau,nU)}{[t_e/U]}$$

$m(\tau)$は、ウインドサイズをτとして、時刻 t が進むに従って変化する $M(t,\tau)$ の平均値である。解析結果より、ウインドサイズを変化させた場合のミッシングページサイズの変化を知ることができる。

スペース・タイム・プロダクト[3] $S(\tau)$

最適なτの値を求めるために使用効率の評価指標としてメモリコストに注目したスペース・タイム・プロダクト $S(\tau)$ を測定した。

$S(\tau)$ は、あるプログラムの実行開始から終了までの占有メモリ量と、占有時間の積であり、次式で表わされる。

$$S(\tau)=\int_{t_0}^{t_e} C(t)\,dt$$

$C(t)$： あるプログラムが、時刻 t において占有しているメモリの大きさ

ここでは、ページフォルトが発生して、主記憶，補助記憶間のページ転送の必要性が生じた時から、転送が完了して再びプログラムの処理が可能となるまでのページ転送時間を考慮して、$S(\tau)$は 次式で表わされる。

$$S(\tau)\fallingdotseq W(\tau)\times(t_{cp}+P_f\times t_r)$$

t_{cp}：あるプログラムを処理するために要したプロセスタイム

P_f：あるプログラムを処理する時に発生したページフォルトの全回数

t_r ：ページ転送時間

4．測定結果

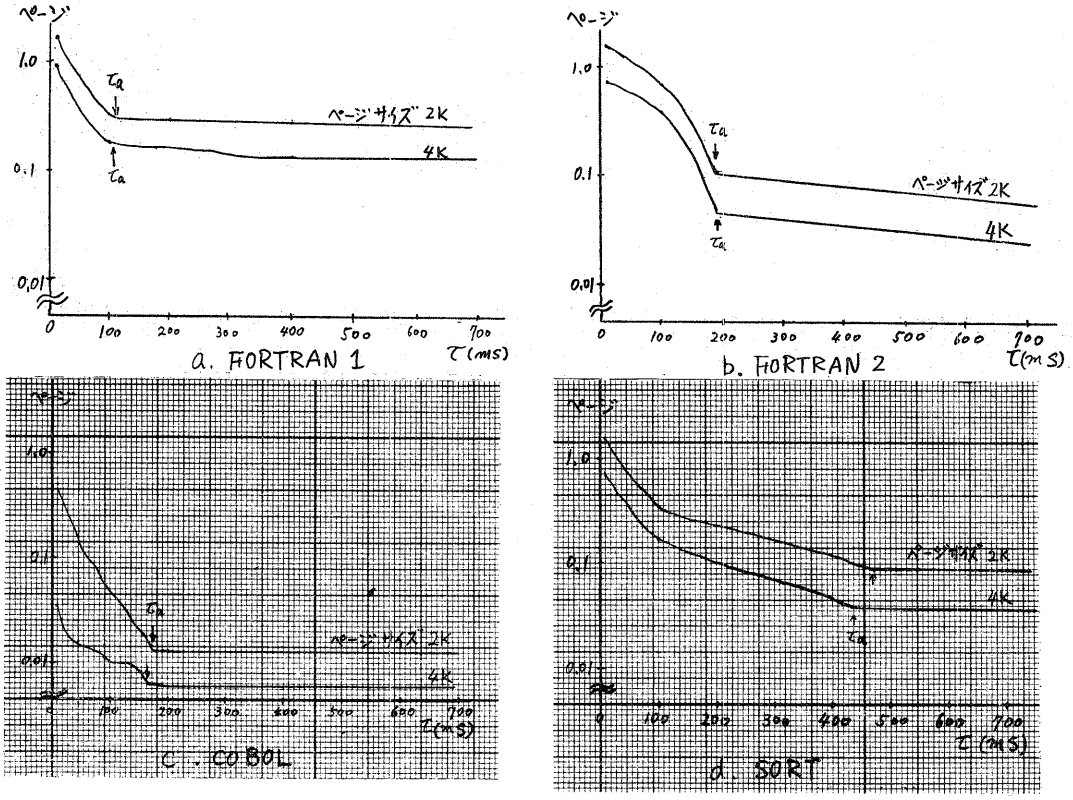
オ5図～オ7図に主な測定結果を示す。

(1) サンプリングタイム

SYDASの機能に従い、メモリアクセス状態を記録するサンプリングタイムとして、1ms単位と10ms単位とを設定した。その結果について比較すると、ほとんど差異は認められない。オ1表より、スペースタイムプロダクトS(τ)の最小値を記録するτ₀の中で、さらに最小の値は、$\tau_0 = 120$ msである。このことから、S(τ)が最小値となるτを検出するには、10msをサンプリングタイムとして設定すれば十分である。

(2) ミッシングページサイズ　$m(\tau)$

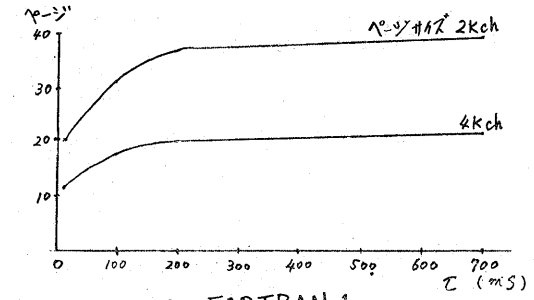
FORTRAN，COBOL，SORTを含め、全ての被測定プログラムについて、$m(\tau)$の曲線は2つの部分に分けることができる。1つは、τが$(0, \tau_a)$の範囲をとるときの曲線であり、曲線の傾きが大きい部分である。他の1つは、τが(τ_a, ∞)の範囲をとるときの曲線であり、傾きが0に近い部分である。FORTRAN及びCOBOLプログラムとSORTプログラムについて比較すると、次のことが明らかである。FORTRAN，COBOLでは$(0, \tau_a)$の間の傾きが大きく、τ_aの存在がはっきりしている。SORTでは$(0, \tau_a)$の間の傾きが小さく、τ_aの存在が前者ほど明確ではない。



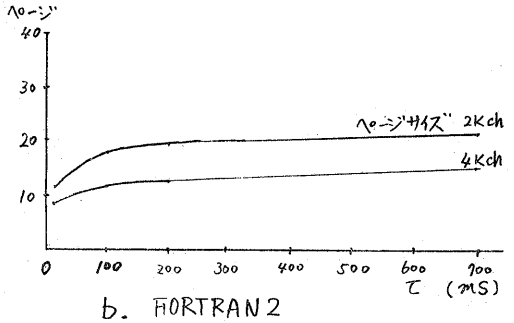
オ5図　ミッシング・ページ・サイズ

(3) 平均ワーキングセットサイズ　W(τ)

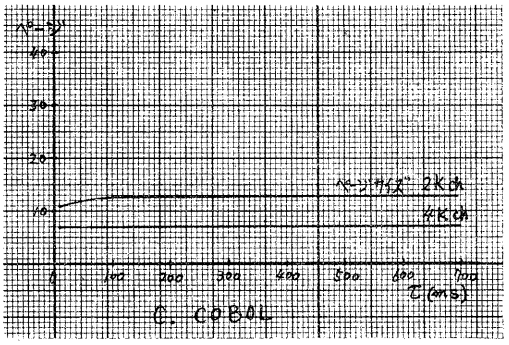
W(τ) の曲線は、τが小さい値において急激に立上り、τの大きい値においては変化が少なく平坦な曲線となる。FORTRAN, COBOLプログラムとSORTプログラムとを比較すると前者のW(τ) の傾きは、τの大きな値において後者における変化率よりも小さい。



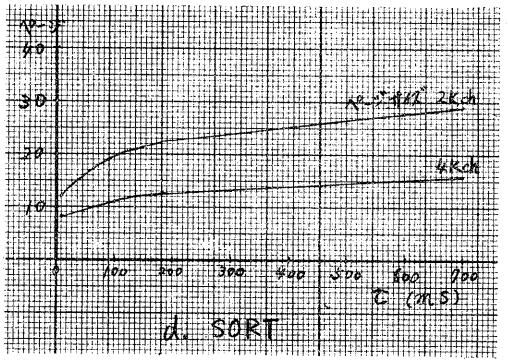
a. FORTRAN 1



b. FORTRAN 2



c. COBOL

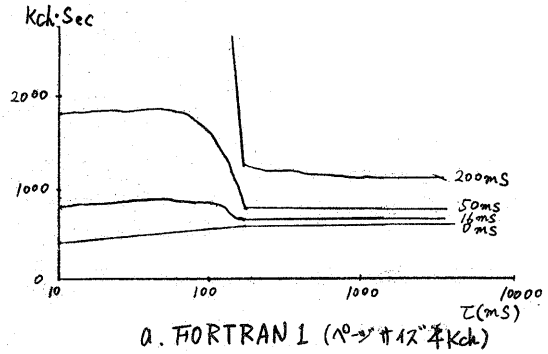


d. SORT

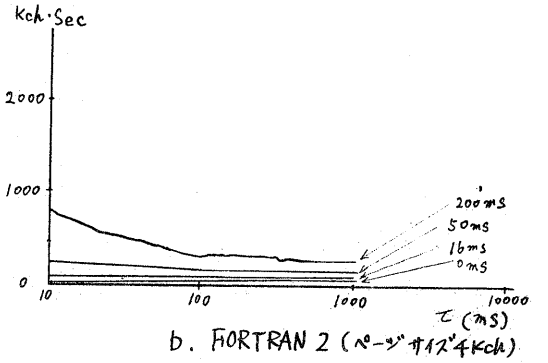
第6図．平均ワーキング・セット・サイズ

(4) スペース・タイム・プロダクト　S(τ)

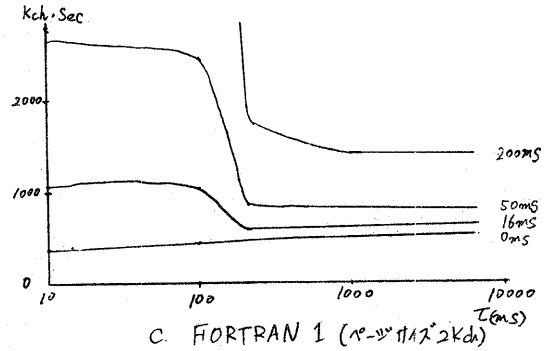
S(τ) の曲線は、τが小さい値では、ページ転送時間が大きいほど極端に大きな値を示す。ページ転送時間が小さいS(τ) では、τによる変化が少ない。転送時間が50 ms以上になると、ほとんどのプログラムでは、τが(0, τa)の範囲で減少しており、τa以後の領域では、FORTRAN, COBOLプログラムではほぼ一定値となるが、SORTプログラムでは、τが大きくなるとS(τ) も大きくなっている。



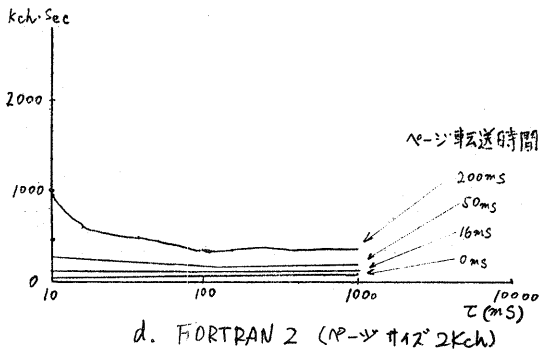
a. FORTRAN 1 (ページサイズ 4Kch)



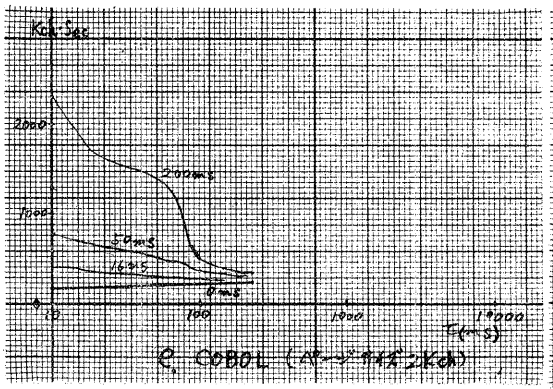
b. FORTRAN 2 (ページサイズ 4Kch)



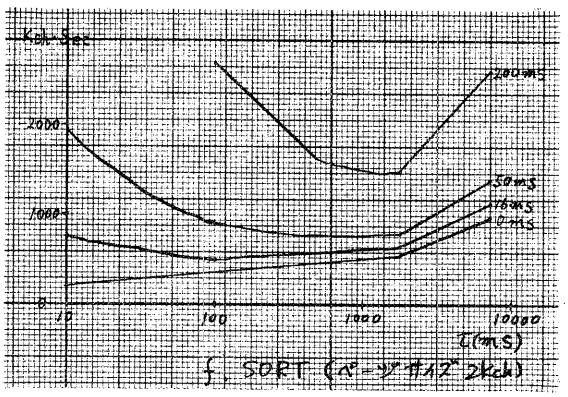
c. FORTRAN 1 (ページサイズ2Kch)



d. FORTRAN 2 (ページサイズ2Kch)



e. COBOL (ページサイズ2Kch)



f. SORT (ページサイズ2Kch)

第7図　スペース・タイム・プロダクト

5. 考察

第1表では、被測定プログラムについて重要と考えられる項目について整理した結果を示す。スペース・タイム・プロダクトについては、ページ転送時間を50mSとした場合のデータである。

(1) 第1表より、スペースタイムプロダクトの最小値におけるウインドサイズτ_0の値は、ページサイズ2Kの場合で、106ms～768msまでの範囲にありτ_0に対応する平均ワーキングセットサイズ$w(\tau_0)$は、12.9ページ～38.4ページまでプログラムによって様々な値を示す。また、$w(\tau_0)/P_A$の値が、ページサイズ2Kの場合で0.60～0.99までのばらつきがあることにより、最適なワーキングセットサイズは、プログラムの大きさに単純に比例するものではないことが明らかである。ミッシングページサイズ$m(\tau)$において、$m(\tau_0)$の値がページサイズ2Kの場合で0.012ページ～0.327ページまでの値であり、$w(\tau_0)$と同様、プログラムによるばらつきが大きい。

(2) ミッシングページサイズ$m(\tau)$の曲線から求めたτ_aとτ_0の値は、一致しないが、スペースタイムプロダクトの最小値$S(\tau_0)$との関係が

$$S(\tau) \leq 1.05 \times S(\tau_0)$$

となるτを満足する値の中にτ_aは含まれる。実際のシステムにおいて、メモリ管理を行なう場合には、ミッシングページに関する情報は、他のパラメータと比較して収集し易いので、τ_aを制御変数として使える可能性がある。

第1表　測定データの主要項目一覧表

プログラム名	ページ・サイズ (Kch)	アクセス・ページ数 P_A	S(τ)の最小値 S_{MIN} (Kch·s)	S_{MIN}のときのτ τ_0 (ms)	S(τ)の1.05倍のτ範囲 (ms) *	$m(\tau_0)$ (ページ)	$w(\tau_0)$ (ページ)	$w(\tau_0)/P_A$	τ_a (ms)
FORTRAN 1	2	49	768	490	250～	0.072	38.4	0.78	180
	4	25	725	540	160～	0.035	21.1	0.84	180
FORTRAN 2	2	29	106	120	110～150	0.327	17.6	0.60	120
	4	16	110	410	90～	0.137	14.5	0.90	120
COBOL	2	13	286	250	160～	0.012	12.9	0.99	170
	4	7	302	160	20～	0.006	7.0	1.00	160
SORT	2	41	610	1130	470～1900	0.051	29.6	0.72	480
	4	21	572	430	330～2000	0.043	14.3	0.68	440

P_A：実行中アクセスされたページの種類の数
S_{MIN}：スペースタイムプロダクト S(τ)の最小値
τ_0：S_{MIN}におけるτの値
＊：$S(\tau) \leq 1.05\, S_{MIN}$ を満足する τ の範囲
$m(\tau_0)$：τ_0におけるミッシングページサイズの値
$w(\tau_0)$：τ_0におけるワーキングセットサイズの値
τ_a：ミッシングページサイズ $m(\tau)$の曲線が大きく変化する部分と比較的平坦な部分の境界点におけるτの値

6　おわりに

ワーキングセットモデルの各解析項目における実測結果から、プログラムが示す特性には、プログラムによってばらつきがあることが明らかとなった。従って、メモリ管理は、実行されるプログラム全体にわたった統一された管理より、むしろ各プログラム毎に管理することが望ましいと考えられる。現実のシステムにおいて、このような制御を行う場合には、制御の複雑化に伴って、導入のためのコストについて検討する必要がある。今後、ここで得たデータを基に、制御のためのアルゴリズムを設定し、評価すると共に、システムへの導入等について検討する予定である。

謝　辞

最後に、日頃御指導頂く藤野部長、林津課長ならびに関連グループの方々に深謝致します。

参考文献

[1] J. Rodriguez - Rosell : Empirical Working Set Behavior
C. ACM Vol. 16 NO. 9 . pp 556 - 560 (1973)

[2] 小野 他 : プログラム動特性の測定
情報処理学会第15回大会(1974), 予稿集 NO. 47

[3] W. W. Chu and H. Opderbeck : Performance of Replacement Algorithms with Different Page Sizes.
Computer . 1975 (11) , pp 14 - 21

[4] P. J. Denning : The Working Set Model for Program Behavior
C. ACM , Vol. 11 . NO. 5 pp 323 - 333 (1968)

[5] 小野 箱崎 : システム・データ収集装置 - SYDAS
電子通信学会全国大会 (1971) , 予稿集 NO. 1053